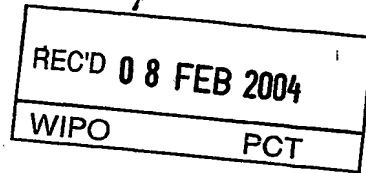


Rec'd PCT/PTO 17 MAY 2005
#2
BUNDESREPUBLIK DEUTSCHLAND



DE 03/3848



**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen:

102 54 285.6

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Anmeldetag:

20. November 2002

Anmelder/Inhaber:

ROBERT BOSCH GMBH, 70442 Stuttgart/DE

Bezeichnung:

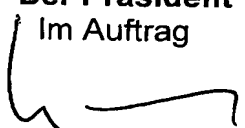
Gateway-Einheit zur Verbindung von Subnetzen,
insbesondere in Fahrzeugen

IPC:

H 04 L 12/66

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 8. Januar 2004
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag





20.11.02 Bea/Pz

ROBERT BOSCH GMBH, 70442 Stuttgart

Gateway-Einheit zur Verbindung von Subnetzen, insbesondere in Fahrzeugen

Stand der Technik

Die Erfindung betrifft eine Gateway-Einheit zur Verbindung von Subnetzen, insbesondere in Fahrzeugen.

Um neue Dienste im Fahrzeug zu ermöglichen ist eine Kommunikation der Steuereinheiten, die sich in unterschiedlichen Bussegmenten befinden, zwingend erforderlich. Eine solche Kommunikation kann nur stattfinden, wenn die unterschiedlichen Bussegmente über eine Gateway-Einheit bzw. Gateway-Einheiten miteinander verbunden sind. Eine Gateway-Einheit, die zwei Bussegmente miteinander verbindet, hat dabei die Aufgabe, Nachrichten, die auf einem Bussegment empfangen werden, auf ein anderes Bussegment weiterzuleiten (Routing). Die Komplexität einer solchen Gateway-Einheit steigt mit der Anzahl der zu koppelnden Bussegmente. Bei der Auslegung der Netzkoppelarchitektur eines Fahrzeugs wird versucht, ein Optimum für die Eigenschaften der tolerierbaren Verzögerung für das Nachrichten-Routing, der Fehlertoleranz, der Flexibilität, der Erweiterbarkeit und der Kosten zu finden. Je nach Anwendung werden zentrale Gateway-Einheiten mit einer Sternarchitektur oder mehrere Gateway-Einheiten, die z.B. mit einem Backbone-Bus verbunden sein können, eingesetzt. Die Gateway-Einheiten, über die unterschiedliche Busse gekoppelt sind, sind konfiguriert (z.B. über Tabellen). Das heißt für das reine Nachrichten-Routing auf ein anderes Segment ist somit keine Abänderung der Software erforderlich. Sobald sich aber die Art und die Anzahl der gekoppelten Bussegmente ändern, ist ein erheblicher Änderungsaufwand erforderlich, bei dem nicht nur die bestehenden Konfigurationstabellen angepasst werden müssen, sondern die komplette Software

umgeschrieben werden muss, um den neuen Anforderungen gerecht zu werden. Die Komplexität der zentralen Konfiguration und der Routing-Software steigt also mit der Anzahl der gekoppelten Bussegmente enorm an.

Vorteile der Erfindung

Durch den modularen Aufbau einer Gateway-Einheit, bei welcher ein in Software ausgebildetes Gateway (logisches Software-Gateway) für das Routing von Nachrichten zwischen genau zwei Subnetzen zuständig ist, wird es ermöglicht, Gateways zu erweitern, ohne dass eine Änderung in der vorhandenen Software des Gateways und/oder in den vorhandenen Konfigurationstabellen erforderlich ist. Durch das Hinzufügen oder Weglassen eines solchen modularen Gateways bei Änderung der Netztopologie werden solche Änderungsmaßnahmen vermieden. Entsprechend ist es möglich, bei einer zentralen Gateway-Einheit ein Bussegment zu entfernen ohne bestehende Kopplungspfade zu beeinflussen.

Ferner wird durch die oben genannten modularen Gateways eine Fehlerbegrenzung erreicht, da bei nicht funktionsfähigem Gateway die anderen Gateways unabhängig davon weiterhin ihre Aufgabe erfüllen. Ein Fehler wird also auf das unmittelbar betroffene Gateway begrenzt, die Kopplung zu anderen Bussegmenten ist nicht berührt. Kommt es auf einem Bussegment zu einem Fehler, so werden die Nachrichten über die anderen Bussegmente ohne Einschränkung weiter geroutet.

Ferner ist das oben skizzierte Konzept flexibel erweiterbar und passt sich der Netzkoppelarchitektur an. Wird ein zusätzliches Subnetz einem zentralen Gateway hinzugefügt, so müssen lediglich die zusätzlichen modularen Gateways hinzugefügt werden. Die bestehenden modularen Gateways sind davon nicht betroffen. Wird ein Subnetz entfernt, werden die Gateways, die auf dieses Subnetz koppeln, entfernt, die restlichen bleiben unverändert bestehen.

Da ferner in einem logischen Gateway immer nur die Funktion enthalten ist, die für das Koppeln der zwei unterschiedlichen Subnetze erforderlich ist, wird eine Nachricht schnellst möglich geroutet. Unnötiger Overhead für das Routing einer Nachricht wird so vermieden. Da ein logisches Gateway genau für die Kopplung von zwei Subnetzen zuständig ist, kann ferner der Informationsfluss gesondert kontrolliert werden, wenn der

Datentransfer zwischen einem Subnetz mit sicherheitskritischen Funktionen und einem Subnetz mit nicht sicherheitskritischen Funktionen stattfindet. Es wird somit also eine optimale Kontrollmöglichkeit für eine Firewall-Funktionalität bereitgestellt. Diese kann jeden Kopplungspfad individuell kontrollieren. So kann ein logisches Gateway, das Nachrichten über eine Luftschnittstelle routet, zur Abwehr von externen Bedrohungen striktere Sicherheitsmechanismen implementieren als ein logisches Software-Gateway, das Nachrichten zwischen zwei CAN-Subnetzen im Fahrzeug routet und direkt keiner externen Bedrohung ausgesetzt ist.

Ferner ermöglicht die dargestellte Architektur ein individuelles Aktiv- und/oder Inaktivschalten eines einzelnen oder mehrerer Gateways, so dass auf diese Weise abhängig von Systemzuständen ein oder mehrere Gateways ein- bzw. ausgeschaltet werden können.

Ferner erfolgt eine Reduzierung der Komplexität der gesamten Gateway-Einheit, da die einzelnen logischen Gateways untereinander überhaupt nicht miteinander verknüpft sind. Es spielt keine Rolle, ob z.B. drei logische Gateways auf einem zentralen Gateway oder auf drei getrennten Punkt-zu-Punkt-Gateways ablaufen.

Weitere Vorteile ergeben sich aus einer konkreten Ausgestaltung der logischen Gateways, nach der in jedem Gateway Routingtabellen vorgesehen sind, über die das Routing der Nachrichten abgewickelt wird und die unabhängig von der Software des Gateways sind. Durch diesen tabellenbasierten Ansatz wird es ermöglicht, ein Tool zur Konfiguration der Gateway-Software einzusetzen. Dieser Ansatz führt auch in vorteilhafter Weise zu einer Priorisierungsmöglichkeit der Nachrichten, so dass bestimmte Nachrichten, die bevorzugt geroutet werden sollen, eine höhere Priorität zugeordnet werden kann als anderen Nachrichten.

Ferner ist in vorteilhafter Weise ein Scheduler vorgesehen, der trotz der Aufteilung in mehrere modulare Software-Gateways gewährleistet, dass die Reihenfolge des Nachrichten-Routings eingehalten wird. Auf diese Weise kann eine Nachricht, die zuerst im Gateway angekommen ist, das Gateway auch zuerst wieder verlassen.

Weitere Vorteile ergeben sich aus der nachfolgenden Beschreibung von Ausführungsbeispielen bzw. aus den abhängigen Patentansprüchen.

Zeichnung

Die Erfindung wird nachstehend anhand der in der Zeichnung dargestellten Ausführungsbeispielen näher erläutert.

Figur 1 zeigt das Grundprinzip der beschriebenen Architektur einer Gateway-Einheit, welche drei Bussegmente miteinander verbindet.

Figur 2 zeigt ein bevorzugtes Ausführungsbeispiel eines solchen Gateways, welches Nachrichten zwischen einem Lowspeed-CAN, einem Highspeed-CAN und einem SPI-Bus vermittelt.

Figur 3 zeigt eine zentrale Gateway-Einheit zur Kopplung zwischen vier Subnetzen, während in

Figur 4 Punkt-zu-Punkt-Gateway-Einheiten zur Kopplung der vier Subnetze mit Hilfe der beschriebenen Gateway-Architektur dargestellt sind.

Figur 5 schließlich zeigt ein in eine Steuereinheit integriertes Gateway anhand eines Schichtenmodells.

Beschreibung von Ausführungsbeispielen

In Figur 1 ist eine Gateway-Einheit 10 dargestellt, an die drei Bussegmente 1, 2, 3 angekoppelt sind und die die Aufgabe hat, Nachrichten von einem Bussegment auf eines der anderen oder beide andere Bussegmente zu verbinden (routen). Grundprinzip der dargestellten Architektur sind die modularanb Gateways (logischen Software-Gateways) 12, 13, 23, wobei ein solches Gateway für das Routing von Nachrichten zwischen genau zwei Subnetzen zuständig ist. Somit routet das Gateway 12 Nachrichten von 1 nach 2 und umgekehrt, das Gateway 13 Nachrichten von 1 nach 3 und umgekehrt, das Gateway 23 Nachrichten von 2 nach 3 und umgekehrt. Jedes logische Software-Gateway beschreibt daher einen individuellen Koppelpfad zwischen zwei Subnetzen bzw. Bussegmenten. Die Gateways 12, 13, 23 sind dabei als Software Programme ausgestaltet, mit deren Hilfe die protokollspezifischen Anpassungen vorgenommen werden, die für ein Nachrichten-Routing zwischen den beiden Subnetzen erforderlich sind. Je nach Ausführungsbeispiel handelt es sich bei jedem Subnetz um ein individuelles Übertragungsmedium. Subnetz 1 kann z.B. ein Lowspeed-CAN, Subnetz 2 ein Highspeed-CAN und Subnetz 3 ein SPI-Bus sein. Kommt ein neues Subnetz hinzu, beispielsweise ein MOST-Bus, so werden zusätzliche logische Software-Gateways integriert. Die Bestehenden müssen nicht

geändert werden. Wird ein Subnetz entfernt, beispielsweise der SPI-Bus, so werden die logischen Software-Gateways 13 und 23 entfernt. Um eine universelle Gateway-Funktion zu haben, sind logische Software-Gateways für alle Kopplungsmöglichkeiten zu schreiben. Diese werden dann je nach Auslegung des zu realisierenden Gateways zum Gesamtsystem kombiniert. Im Regelfall wird es so sein, dass nicht alle Subnetze direkt miteinander gekoppelt sind, so dass nur ausgewählte Kopplungspfade mit ausgewählten logischen Software-Gateways zu versehen sind. Soll jedes Subnetz mit jedem anderen gekoppelt werden, so benötigt man $N*(N-1)/2$ logische Software-Gateways. Die Variable N entspricht dabei der Anzahl der im Gesamtsystem vorhandenen Subnetze. Für drei Subnetze ergeben sich damit 3 logische Software-Gateways, für vier Subnetze 6 und für fünf Subnetze 10. Ob sich diese logischen Software-Gateways in einem zentralen Gateway befinden oder in mehreren verteilten Punkt-zu-Punkt-Gateways spielt eine untergeordnete Rolle.

Figur 2 zeigt ein detaillierteres Ausführungsbeispiel der Grundzüge der modularen Gateway-Architektur gemäß Figur 1. Das Gateway 10, welches vorzugsweise als Programm in einem Mikrocontroller einer Steuereinheit realisiert ist, umfasst neben den dargestellten modularen Software-Gateways (:CANCAN, :CANSPI) busspezifische Sendeeinheiten, die den Zugriff auf das Busmedium kontrollieren. Jedem Bussegment sind Empfangsobjekte zugeordnet (:Rx-CAN, :Rx-SPI), welche die Entscheidung treffen, in welches logische Software-Gateway eine ankommende Nachricht geroutet wird. Entsprechend gibt es für den Sendevorgang busspezifische Sendeobjekte (:TxCAN, :TxSPI), die den Zugriff auf den jeweiligen Bus kontrollieren und verhindern, dass mehrere modulare Software-Gateways das Sendemedium gleichzeitig belegen.

Die Software-Gateways (in Figur 2: CANCAN:CANSPI) setzen sich intern aus mehreren Software-Objekten zusammen, die ankommende Nachrichten zwischenspeichern und die protokollspezifischen Anpassungen vornehmen. Eine einfache Anpassung ist z.B., dass eine CAN-Nachricht vom Highspeed-CAN mit der ID 100 auf den Lowspeed-CAN mit der ID 200 gesendet werden muss. Diese protokollspezifischen Anpassungen werden dann durch entsprechende Programme (z.B. im einfachsten Fall durch eine Tabelle) vorgenommen. Für die protokollspezifischen Anpassungen, die innerhalb der logischen Software-Gateways durchgeführt werden, kommen Konfigurationstabellen zum Einsatz.

In einer bevorzugten Ausführung werden die busspezifischen Empfangsobjekte über sogenannte Routingtabellen konfiguriert, mit deren Hilfe entschieden wird, ob eine ankommende Nachricht an kein, an ein oder an beide logischen Software-Gateways weitergegeben wird. In der Routing-Tabelle werden somit für jeden ankommenden Nachrichtentyp die Weiterbehandlung der Nachricht abgelegt. Ferner kann durch unterschiedliche Schnelligkeit der Busse es vorkommen, dass von einem zum anderen Bussegment nur jede 5. Nachricht eines bestimmten Typs (z.B. Motordrehzahl) weitergeleitet wird. Auch dies kann mittels der genannten Routingtabellen im Empfangsobjekt realisiert sein. Diese Routingtabellen sind dabei unabhängig vom Source-Code des eigentlichen Gateways, so dass eine Veränderung der Routingtabellen nicht oder nur unwesentliche Änderungen der Software des betroffenen modularen Gateways nach sich zieht. Die busspezifische Empfangseinheit sucht die gefundene Nachricht in der Routing-Tabelle auf und entscheidet aufgrund der darin enthaltenen Informationen, welches logische Software-Gateway die Nachricht zur Weiterverarbeitung erhält.

Die busspezifischen Sendeeinheiten, bzw. die dort vorgesehenen Programme, kontrollieren den Zugriff auf den Bus. Ist der Bus gerade belegt, so stellen sie sicher, dass von keinem logischen Software-Gateway gesendet wird.

Zusätzlich puffern wie oben erwähnt die logischen Software-Gateways die Nachrichten zwischen, um einen Nachrichtenverlust zu verhindern, z.B. dann, wenn das Bussegment auf dem gesendet werden soll, gerade belegt ist. Eine Nachricht wird also vor ihrer direkten Weiterleitung in einer Warteschleife abgelegt. Das Ablegen einer Nachricht in dieser Warteschleife wird beim internen Scheduler der Gateway-Einheit vermerkt. Dieser veranlasst dann das Senden der Nachricht, indem er eine Nachricht an das entsprechende modulare logische Software-Gateway sendet. Dieses veranlasst dann das Versenden der Nachricht. Will also ein logisches Software-Gateway eine Nachricht senden, so muss es diesen Sendewunsch beim Scheduler anmelden. Von der Reihenfolge der Anmeldung ist es abhängig, welches Software-Gateway zuerst die Berechtigung für das Senden einer Nachricht erhält. Dieses Konzept stellt die Einhaltung der Reihenfolge der Nachrichten sicher. Gibt es besondere hochpriorisierte Nachrichten im System, so stellt der Scheduler mehrere Methoden bereit, welche von den logischen Software-Gateways aufgerufen werden können, um einen Sendewunsch anzumelden. Der Scheduler arbeitet dann immer zuerst die hochpriorisierten Anforderungen ab und dann die normalen und erteilt

prioritätenabhängig Sendeberechtigungen an die logischen Software-Gateways.
Beispielsweise ist jede Sendewunsch mit einem die Priorität der Nachricht
repräsentierenden Datum versehen oder der Scheduler umfasst eine Tabelle, in der die
Prioritäten der Nachrichten vermerkt ist, aus der der Scheduler die Priorität ausliest.

Durch die oben dargestellte Architektur und Vorgehensweise ist man in der Lage, die
Gateway-Einheit über Tabellen zu konfigurieren, ohne deren Software selbst abzuändern.
Beispielsweise kann über eine Abänderung der Parametersätze im Speicher das Gateway
für ein anderes Nachrichten-Routing umprogrammiert werden. Werden die gleichen
Schnittstellen verwendet, so lässt sich die Gateway-Software ausschließlich über
Parametersätze konfigurieren. Bei dem Anschluss anderer Schnittstellen an das Gateway,
ist ein modularer Software-Baustein in das Gateway zu integrieren. Somit werden
unterschiedliche Gateway-Konfigurationen durch das Zusammenführen von Software-
Modulen beispielsweise aus Bibliotheken und aus dem Bereitstellen der Routing-
Information generiert. Die Neuintegration einer CAN-Schnittstelle mit neuer CAN-
Matrix beschränkt sich im wesentlichen auf das Eingeben der neuen Routing-Information
in die Routing-Tabelle. So können CAN-CAN-Gateways unterschiedlicher Baudraten in
kürzester Zeit in ein System eingepasst werden. Die Testbarkeit und Verifizierung des
entstehenden Codes werden dadurch vereinfacht, weil der konfigurationsunabhängige
Code zentral getestet wird und neben dem Systemtest ausschliesslich ein Integrationstest
für das neue logische SW-Gateway bzw. die neue Konfiguration durchgeführt werden
muß. }

Figur 3 zeigt ein Gateway 10 zur Kopplung 4 unterschiedlicher Bussegmente, einen
Low-speed-CAN, einen Highspeed-CAN, ein SPI-Bus und einen MOST-Bus. Die
vorstehend beschriebene Architektur ist auch hier eingesetzt, wobei logische Software-
Gateways (:CAN-MOST, :CAN-SPI, :CAN-CAN, :SPI-MOST) eingesetzt werden, die
jeweils einen individuellen Kopplungspfad realisieren. Daneben sind busspezifisch
Empfängermodule (:Rx-Most, :Rx-CAN, :Rx-SPI) und Sendemodule (:Tx-Most, :Tx-
CAN, :Tx-SPI) wie vorstehend beschrieben dargestellt. Die dargestellte Architektur zeigt
eine zentrale Gateway-Einheit, welche die genannten vier Bussegmente miteinander
verbindet. In Figur 4 ist eine andere Netzwerktopologie dargestellt, welche sechs Punkt-
zu-Punkt-Gateway-Einheiten 10a, 10b, 10c, 10d, 10e und 10f aufweist. Jedes dieser
Punkt-zu-Punkt-Gateways enthält die oben dargestellte logische Software-Gateway-
Struktur mit Sende- und Empfängermodulen zur busspezifischen Anbindung. Es zeigt

sich, dass durch die oben dargestellte Architektur die physikalische Netzkoppelarchitektur zwischen den beiden Extremen zentrales Gateway und Punkt-zu-Punkt-Gateway alle denkbaren Mischformen enthalten kann. Die Software-Architektur ist unabhängig von der physikalischen Netzkoppelarchitektur, so dass sie die Kopplung in allen denkbaren Architekturen erlaubt. Unterschiedlich kann sein, dass bei der zentralen Gateway-Variante die Software auf einem Mikrocontroller, in der dezentralen Variante auf unterschiedlichen Controllern abläuft.

Zur Konfiguration der Gateway-Einheit ergeben sich verschiedene Möglichkeiten. Die Konfiguration der Routing-Entscheidung erfolgt über Routing-Tabellen. In diesem Fall entscheiden die buspezifischen Empfangsobjekte, in welche logischen Software-Gateways eine Nachricht weiterzuleiten ist. Diese Empfangsobjekte werden deshalb mit Routing-Tabellen konfiguriert, welche angeben, welche Nachrichten in welches Subnetz und ggf. unter welchen Randbedingungen (z.B. jede 5., etc.) weiterzuleiten sind. Die Software der Software-Gateways realisiert dann die buspezifischen Anpassungen und ist von dem eigentlichen Routing-Vorgang unabhängig. Die Konfiguration der Software-Gateways erfolgt dann durch Anpassung von Protokollparametern. In diesem Fall werden die logischen Gateways über Tabellen konfiguriert, die angeben, wie die Protokoll-Parameter umzusetzen sind. Hier kann konfiguriert werden, dass eine Nachricht mit dem Identifikationscode 100 bei dem Versenden auf dem anderen Netzsegment die Identifikationsnummer 200 tragen muss. Ferner sind, um die Gateway-Software an die unterschiedlichen Netzkoppelarchitekturen anzupassen, die Routing-Tabellen, mit denen die buspezifischen Empfangsobjekte konfiguriert werden, aufzuteilen bzw. zusammenzuführen. Diese Aufgabe wird durch einen internen Scheduler erledigt, welcher die logischen Software-Gateways in einer zentralen Gateway-Einheit untereinander koordiniert. Der Scheduler muss für die verschiedenen Gateway-Varianten individuell generiert werden.

In einer anderen Ausführungsform, die in Figur 3 dargestellt ist, handelt es sich bei dem Gateway um kein Standalone-Gateway, sondern um ein Gateway, das in ein Steuergerät mit zusätzlichen Anwendungsfunktionen integriert wird. Dort kann die Gateway-Software auch die Funktionalität eines normalen Kommunikationsdecks übernehmen. Das heißt, es muss hier auch möglich sein, Nachrichten zu den eigentlichen Anwendungen weiterzuleiten und Nachrichten zum Versenden von diesen entgegenzunehmen. Hierzu werden weitere Objekte benötigt, welche die Fähigkeit

besitzen, die schichtspezifischen Protokollparameter zu entfernen bzw. hinzuzufügen, um die Nachricht in die nächsthöhere bzw. nächstniedrigere Schicht weiterzureichen. Diese zusätzlichen Objekte sind in der Regel Bestandteil der Software des normalen Kommunikationsnetz. Figur 5 zeigt das Schichtenmodell eines Steuergeräts 100, in welchem ein CAN-CAN-Gateway integriert ist. Dabei wird das Anwendungssystem I und das Kommunikationssystem II unterschieden. Es sind drei Schichten 1, 2, 3 abgebildet, wobei in einer ersten Schicht ein Treiber 102 für den Lowspeed-CAN und ein Treiber 104 für den Highspeed-CAN vorgesehen sind. Ferner wurden in der Vermittlungsschicht 3 zusätzliche Objekte eingefügt (CAN-Schicht 3), die über die Empfangs- und Sendeobjekte Rx3 und Tx3 mit den Anwendungen A, B und C kommunizieren. Diese zusätzlichen Objekte führen, falls erforderlich, eine Zwischenpufferung der Nachrichten durch und fügen protokollspezifische Parameter hinzu oder entfernen diese. Das logische Software-Gateway, das in diese Schicht integriert ist (CAN-CAN) routet Nachrichten vom einen Bus zum anderen. Die Empfangs- und Sendeobjekte Rx2 und Tx2 werden entsprechend dem vorstehend Beschriebenen für das Routing von Nachrichten zwischen den beiden CAN-Bussen verwendet. Sie stellen die Schnittstellen zwischen den Schichten dar. In einer weiteren Ausführung ist durch entsprechende Erweiterung es möglich, zwei Subnetze auf unterschiedlichen Schichten zu koppeln. Dies ist beispielsweise dann erforderlich, wenn ein Transport-Protokoll (z.B. ISO TP) zum Einsatz kommt. In diesem Fall existiert dann pro Schicht, in der eine Kopplung stattfindet, ein logisches Software-Gateway. So kann beispielsweise ein CAN-CAN-Gateway in der Schicht 3 vorgesehen sein, welches CAN-Nachrichten (z.B. Geschwindigkeitsinformation oder Tankfüllstand) in der Schicht 3 transportieren während ein weitere CAN-CAN-Gateway in einer Schicht 4 Transportinformationen koppelt, welche beispielsweise einen Text für die Anzeige im Fahrzeugcomputer enthält. Eine Kopplung in höheren Schichten kann auch erforderlich sein, um den Inhalt einer Nachricht die weitergeleitet werden soll überprüfen zu können. Der Inhalt kann nur dann analysiert werden, wenn die komplette Nachricht empfangen ist.

19.11.02 Bee/Pz

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Patentansprüche

15

20

30

35

1. Gateway-Einheit zur Verbindung von Subnetzen, insbesondere in Fahrzeugen, wobei die Gateway-Einheit wenigstens zwei Subsysteme miteinander verbindet, dadurch gekennzeichnet, dass die Gateway-Einheit aus wenigstens einem modularen Software-Gateway besteht, welches Nachrichten zwischen genau zwei Subnetzen routet.
2. Gateway-Einheit nach Anspruch 1, dadurch gekennzeichnet, dass wenigstens drei Subnetze an die Gateway-Einheit angeschlossen sind, wobei modulare Software-Gateways eingesetzt werden, wobei jedes modulare Software-Gateway Nachrichten zwischen genau zwei Subsystemen routet.
3. Gateway-Einheit nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ferner für jedes Subnetz busspezifische Empfangsobjekte vorgesehen sind, welche die ankommenden Nachrichten an ausgewählte Software-Gateways weitergeben.
4. Gateway-Einheit nach Anspruch 3, dadurch gekennzeichnet, dass die Empfangsobjekte Routing-Tabellen umfassen, in denen die Behandlung ankommender Nachrichten konfiguriert sind.
5. Gateway-Einheit nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ferner für jedes Subnetz busspezifische Sendeobjekte vorgesehen sind, welche den Zugriff auf den jeweiligen Bus kontrollieren.

6. Gateway-Einheit nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass das modulare Software-Gateway ankommende Nachrichten zwischenpuffert und protokollspezifische Anpassungen vornimmt.

- 5 7. Gateway-Einheit nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass bei Integration des Gateways in ein Steuergerät mit Anwendungssystem wenigstens ein modulares logisches Gateway in einer Schicht des Kommunikationssystems vorgesehen sein können, wobei das logische Gateway genau zwei Subsysteme verbindet.

10

19.11.02 Bee/Pz

5

ROBERT BOSCH GMBH, 70442 Stuttgart

10

Gateway-Einheit zur Verbindung von Subnetzen, insbesondere in Fahrzeugen

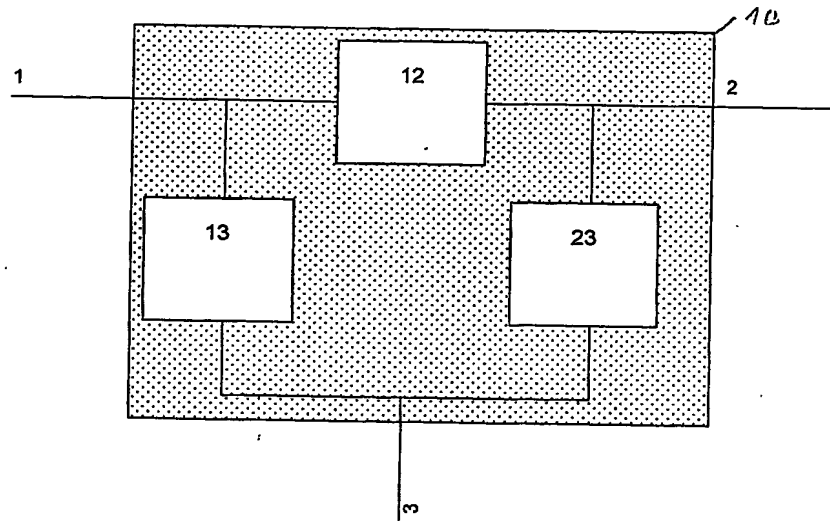
Zusammenfassung

15

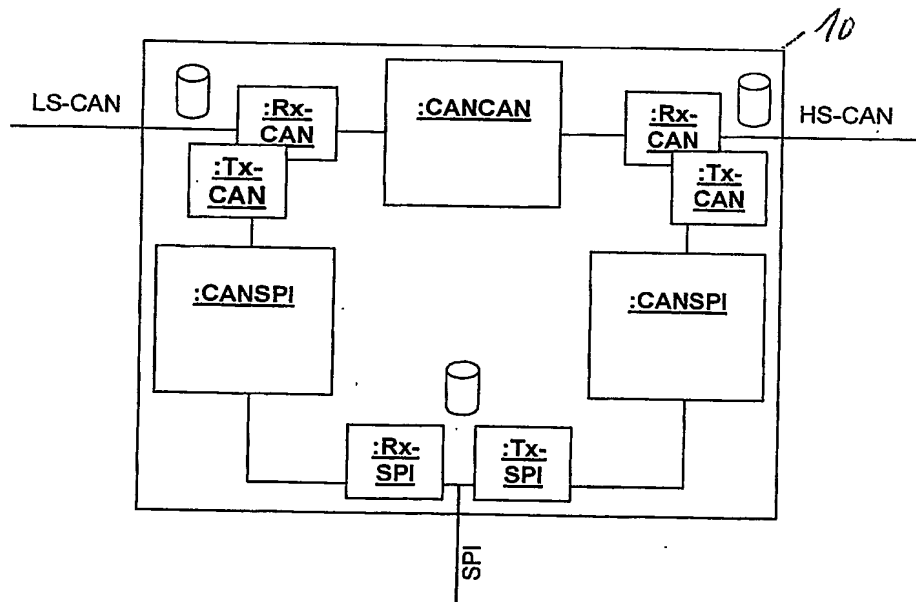
Es wird eine Gateway-Einheit zur Verbindung von Subsystemen, insbesondere in Fahrzeugen vorgeschlagen, bei welchem wenigstens ein modulares logisches Software-Gateway eingesetzt wird, welches Nachrichten zwischen genau zwei Subnetzen routet und auf diese Weise nur einen individuellen Kopplungspfad bereitstellt.

114

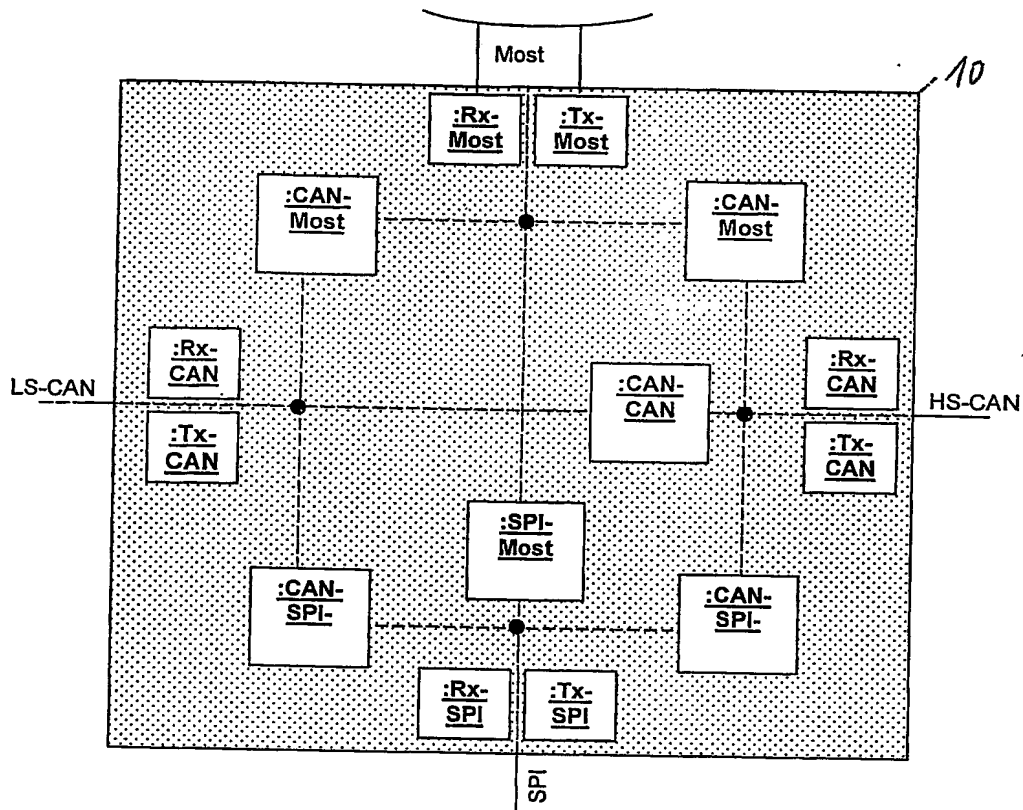
R. 304634



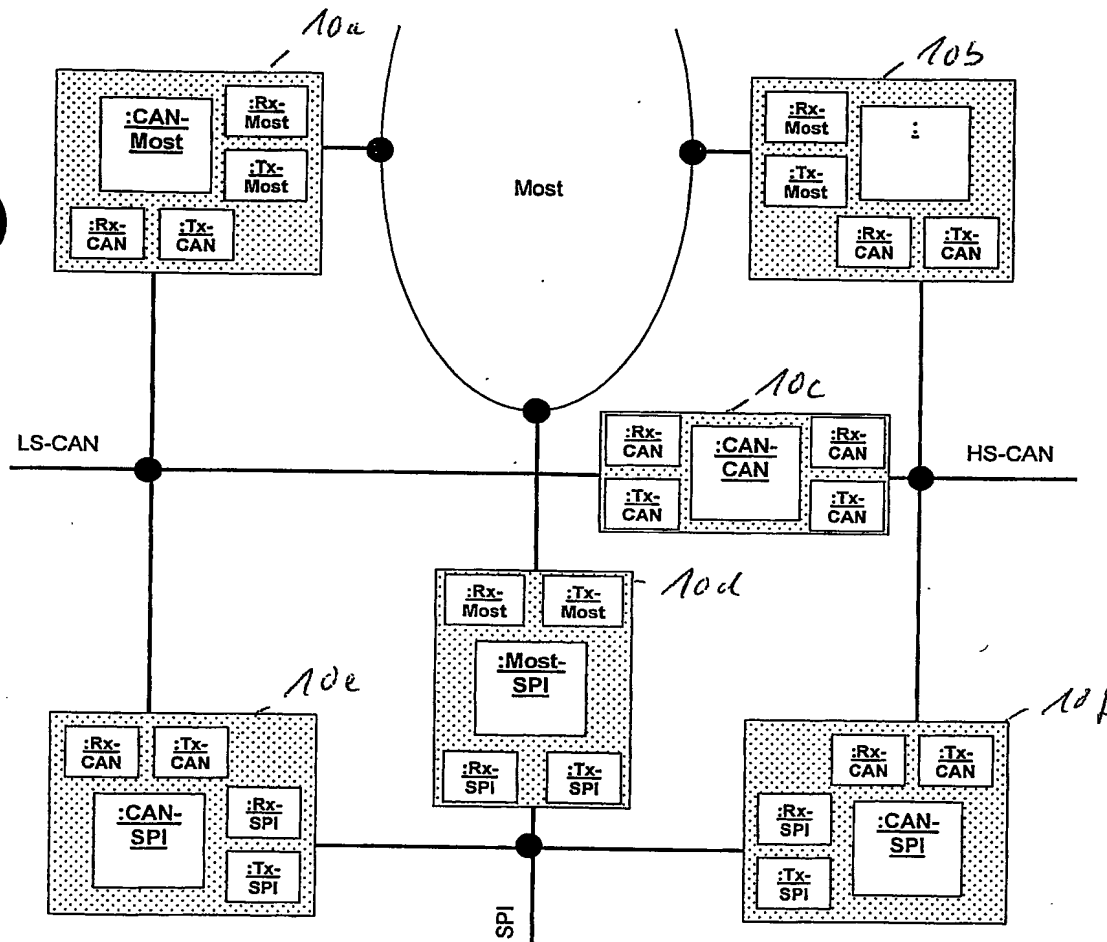
Figur 1



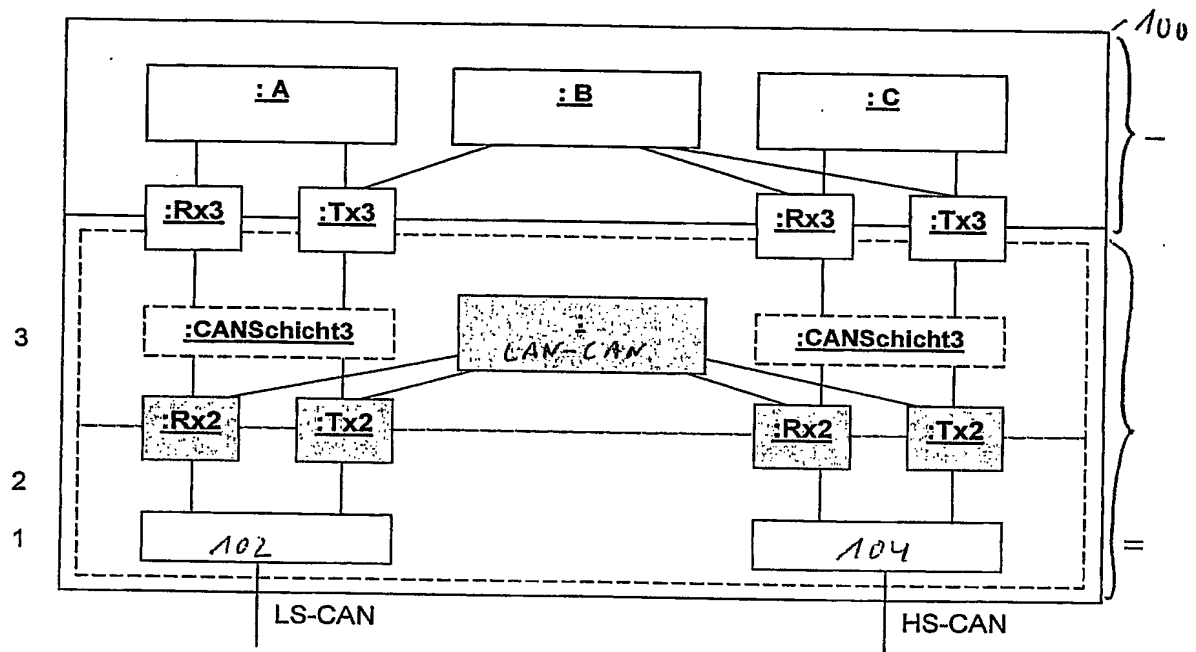
Figur 2



Figur 3



Figur 4



Figur 5

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.